Assignment 2 (Individual Assignment)

Due Date

23 April 2022 17:00 (Saturday)

Declaration of Original Work

Plagiarism is a serious misconduct. No part of students' assignment should be taken from other people's work without giving them credit. All references must be clearly cited. Any plagiarism found in students' completed assignments can lead to disciplinary actions such as mark deduction, disqualification or even expulsion by the College.

____

By submitting this assignment to the subject lecturer through Moodle, you hereby declare that the work in this assignment is **completely** your own work. No part of this assignment is taken from other people's work without giving them credit. All references have been clearly cited.

You understand that an infringement of this declaration leaves you subject to disciplinary actions such as mark deduction, disqualification or even expulsion by the College.

Instruction

Each of you is required to submit a MS Word 2016 file (**with the template provided**) showing your answers, programs and testing screen dumps. Your MS Word file should contain a cover page showing your name, your student ID no, course code (SEHH2239), course name (Data Structures), class (201) and date.

All programs must be written in Python programming language. For each programming-type question, please also write down the above-mentioned particulars as comments on the .py source file, compile and test it with at least three testing cases. Show the program source code and the screen dumps of testing in the MS Word file. Each program must be successfully compiled and can execute, otherwise 50% of the total marks of that question will be deducted.

All submitted assessments will be evaluated with Python version 3.7.12 (the current python version used in Google Colab). Your submitted assessments must run without errors on Google Colab.

Unless otherwise instructed, you **MUST NOT** import any modules in your submitted assessments **Items to be Submitted**

1. **For each student,** you need to convert the MS Word to **PDF** file before submission. Submit a softcopy to Moodle
2. **For each student,** put the MS Word file together with the program files (both .py source files) into a folder (**folder name format:** Student Name_Student ID No e.g**.** ChanTaiMan_16001234A). Use a compression software (e.g. Winzip) to compress the folder

(e.g. ChanTaiMan_16001234A.zip) (***Submit the compressed folder e.g. ChanTaiMan_08001234A.zip via individual class Moodle. In case the Moodle's Assignment Submission System is not available, send the softcopy to me through e-mail***)

Attention:

While submitting the **softcopies** via Moodle, a timestamp will be placed on the softcopies of your assignment. There will be a sharp cut-off time at Moodle, so late assignments will be recorded at Moodle. Softcopies submitted via email or other means will NOT be accepted unless the Moodle is not available. As many students will submit their assignments to Moodle at around the deadline time, it normally takes longer for uploading your assignment, so it is strongly suggested that you start submitting earlier, say at least 45 minutes before the deadline. Marks will be deducted for late submission. Successful submission of this assignment includes the submission of both softcopy and hardcopy. Missing either softcopy or hardcopy is not successful submission.

Plagiarism will be penalized severely. Marks will be deducted for assignments that are plagiarized in whole or in part, regardless of the sources.

Late submission is liable to a **penalty of 10%** of the available marks for **each day late**; Saturdays, Sundays and holidays are counted.

Question 1

Construct the expression tree for the arithmetic expression whose **postfix** representation is

```
a b * c d * +
```

Where a, b, c and d are numbers based on your student ID as follow:

Integers	Based on your student ID number, formed by
a	4th and 5th digits
b	5th and 6th digits
c	6th and 7th digits
d	7th and 8th digits

Then give

a) The prefix representation

b) The infix representation

c) The actual numeric value of the expression

Question 2

A *mailing model* is a messaging pattern where senders sending message to specific receivers. This model usually developed with *queue*.

a) Complete the following class `MyMailQue` according to the comment described:

```
class MyMailQue:
      def __init__(self) :
            # implement a queue with an array

      def enqueue(self, data):
            # insert data at the end of queue

      def dequeue(self):
            # take out data at the beginning of the queue
            # remove it and return the data taken out

      def display(self):
            # show all the data in the queue
```

b) Define a class `Mailbox` which contain a dictionary of `MyMailQue` as the following, complete coding as the comment described.

```
class MailBox:
      def __init__(self):
            # define a dictionary which
            # contain mail queue (MyMailQue).

      def createMailBox(self, addr) :
            # create a new mail queue attached to dictionary defined
            # with addr as key
```

```
        def send(self, sender, receiver, mailbody) :
              # find mail queue of the receiver in dictionary,
              # enqueue a data as tuple which include sender
              # and mailbody to the queue

        def receive(self, addr) :
              # find mail queue in dictionary by addr
              # return the first item in queue with (sender, mailbody)
              # remove it in the queue.
```

c) Execute the following program and show the output.

```
Mailserver = MailBox();
Mailserver.createMailbox(<your mail address>);
Mailserver.createMailbox(<your classmate mail address>);

Mailserver.send(<your mail address>, <your classmate mail address>,
" Hello !  How are you ?" );

Mailserver.send(<your mail address>, <your classmate mail address>,
"Please prepare assignment for Data structure ");

Mailserver.send(<your classmate mail address>, <your mail address>,
"I got your assignment and will prepare it ! ");

Mailserver.send(<your mail address>, <your classmate mail address>,
"Thanks you and see you later ! ");


mail = Mailserver.receive(<your mail address>);
print("My receive mail :" + str(mail) );

mail = Mailserver.receive(<your classmate mail address>);
print("My classmate receive mail :" + str(mail) );
```

Please provide **THREE** testing cases for the program.

Question 3

Fill in the node value of Figure 1 based on your student number as following table.

Integers	Based on your student ID number, formed by
a	4th and 5th digits
b	5th and 6th digits
c	6th and 7th digits
d	7th and 8th digits
e	4th and 6th digits
f	5th and 7th digits
g	6th and 8th digits

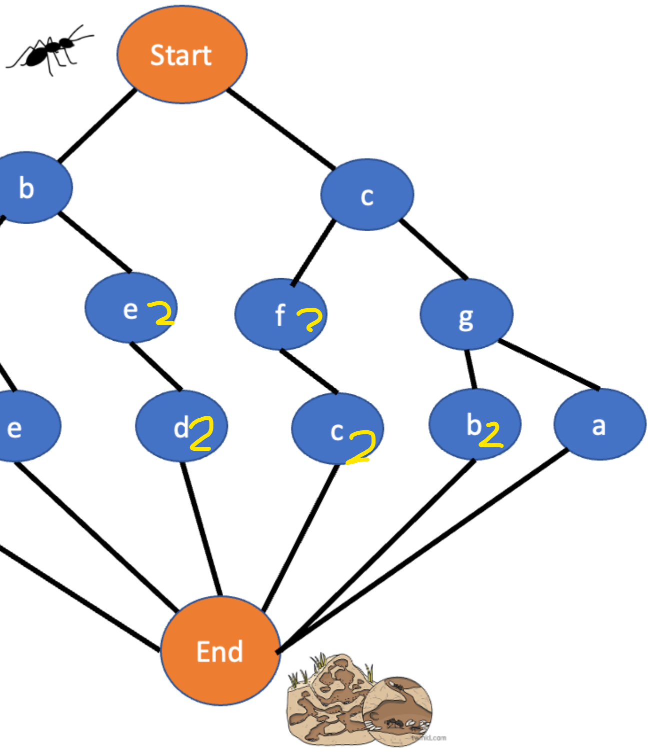

Figure 1

a) Prepare the following two classes:

(i) Define a class `Node` with attribute: `data`, `left_node` and `right_node`.

(ii) Define a class `Path` with a `searchPath` function to to find a path from data="Start" to data="End" as show in the Figure 2. It will return the path from `Start` to `End` such as `Start->b->d->End`.

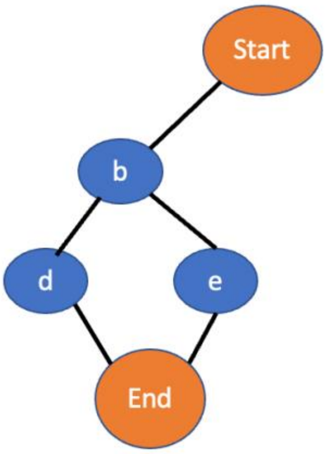

Figure 2

b) In Figure 1, An ant would like to find the path with the smallest total number along the path to its nest. A path is link up of nodes from `Start` to `End`, e.g. for a path `Start -> c -> g -> b -> End`, its total along this path would be `c+g+b`.

Write an additional function, `searchAllPath` for the class `Path` which is to list out all the path that can go from Start to End with its total number based on Figure 1. Sample output as following: only print but not save??? need asking

```
Start->b->d->f->End          10
Start->b->e->d->End          20
...
```

c) Write another find path function `searchPathByWeight` to class `Path` which decision is based on the location of the ant in Figure 1. For example, if the ant is at node `b`, it's next step will based on comparing value at node `d` against `e` and then follow the smallest one.

d) Discuss the advantages and disadvantages of two different methods to find the path with smallest total from `Start` to `End`.

Please provide **TWO** testing cases for the program (Apart from the student ID version) with each case should have more than 30 nodes with different number and its height should not over 10 levels.

Question 4

Considering a **Binary Search Tree** with nodes *a* to *g*, where *a* to *g* are formed based on your student number with following form.

Integers	Based on your student ID number, formed by
a	4th and 5th digits
b	5th and 6th digits
c	6th and 7th digits
d	7th and 8th digits
e	4th and 6th digits
f	5th and 7th digits
g	6th and 8th digits

a) What is the **BEST** structure of your binary search tree for searching? Please draw it out.

b) What is the **WORST** structure of your binary search tree for searching? Please draw it out.

c) What is the best order of your input so as to build the Binary Search Tree as mentioned in (a)?